Development of Lightweight Cryptographic Scheme for Internet of Things

Proposal submitted for the registration of Ph.D. Degree in Computer Science.

Submitted By:

Mr. AbdulRazzaqHasan Abdullah Al-Ahdal
Alahdal201211@Gmail.Com
00919595599780

B.SC ,Hodeiadh University , Hodeiadh- Yemen 2004
M.Sc, The Swami Ramanand Teerth Marathwada University, Nanded, India 2016

Under the Guidance of
Dr. Deshmukh N.K.
(Assistant Professor)

School of Computational Sciences
S.R.T.M University Nanded, Marathwada

1. INTRODUCTION

1.1 Internet of Things

The Internet of Things (IoT) revolution is turning everyday objects into *smart* ones. Such *smart things* are able to interact among themselves and with the environment they are in order to fulfill a given goal [1]. As a result, a global network infrastructure is being created, supporting the provisioning of innovative and customized services to individuals and businesses in different application domains. The resulting system may include an extremely large number of heterogeneous devices, raising integration and scalability challenges to be addressed.

Security & privacy are widely acknowledged to represent critical issues in such a context [2]. On throne hand, the confidentiality and the integrity of the transmitted and stored information has to be guaranteed, and authentication and authorization mechanisms have to be provided to prevent unauthorized users or devices to improperly access the system. On the other hand, privacy of users, understood as ability to support data protection and users anonymity has to be ensured, a critical as pectin particular in the presence of personal and/or sensitive information [3]. Beyond security, also data quality represents an essential requirement for the adoption at scale of IoT services. The provided information should be accurate, timely and complete, since, in some scenarios, errors or missing values might have critical impact on actions or decisions of the IoT system itself [4]. Indeed, as in IoT-enabled services and applications may make use of different data sources, the user (or the application itself) has to be aware of the security and quality level of the data being accessed, in order to take informed decisions about their usage.

As a result, what emerges is the need of a system able to deal with heterogeneous data sources and to evaluate the security and the quality of the information being collected, processed and transmitted, possibly in real time and in an automatic manner. Furthermore, such a system shall be able to work in the absence of a prior complete knowledge of the sources themselves, since IoT environments are highly dynamic and different kinds of attack may occur. In fact, in such a scenario, data may be compromised by rogue devices, hindering the correctness and confidentiality of the information (e.g., data integrity violation, man-in-the-middle attacks, packet sniffing).Source authentication issues (e.g., compromised keys, session violation) shall also be accounted for. In order to

deal with such issues, a mechanism for the assessment of data quality and security is proposed in this paper, supported by a number of novel algorithms aimed at analyzing the data sources as well as the data they generate over time. As far as security assessment is concerned, the presented algorithm is new and, according to the authors' knowledge, the first of its kind. Its aim is to assign a level of robustness to each data source according the following security features: integrity, confidentiality, authentication system, privacy. Therefore, the proposed solution does not tackle the security attacks directly, but aims at minimizing the associated risks by letting users and applications be aware of the security and data quality level.

The proposed solution is integrated in an existing IoT middleware, named *NetwOrked Smart objects (NOS)*[5]. NOS are conceived as computationally powerful devices connected to create a distributed processing and storage layer able to process the data acquired from large-scale IoT deployments close to the actual data sources. NOSs collect the data generated by nearby IoT devices, process them and finally transmit the processed data on a publish/subscribe broker. Such a middleware includes provisioning for users and applications to dynamically specify the level of security and data quality suitable for their own purpose. The distributed architecture automates the deployment of adequate filters for ensuring that only qualified data is being used by the actual service. This represents a clear innovation over conventional one-size-fits-all approaches, which provide the same information to all consumers, often without considering his/her requirements in terms of security, privacy and data quality.

The algorithms integrated within the prototype are validated in an experimental setup characterized by the usage of real-time open data feeds and, contextually, assessing the resulting security and quality level. With respect to the original highly modular and lightweight prototype presented in [5], NOS functionalities are significantly improved in this work, enhancements including:(i) a set of methods for the distributed and autonomic management and run-time optimization of the middleware platform; (ii) a set of secure and privacy aware mechanisms and their integration in the proposed platform; (iii) a set of data quality assessment methods that can be used for different types of data and indifferent scenarios; (iv) standardized interfaces and data models for applications/services to access qualified IoT information following a publish/subscribe

architecture, where raw data are enriched with metadata specifying their security and quality levels.

1.2 Architecture

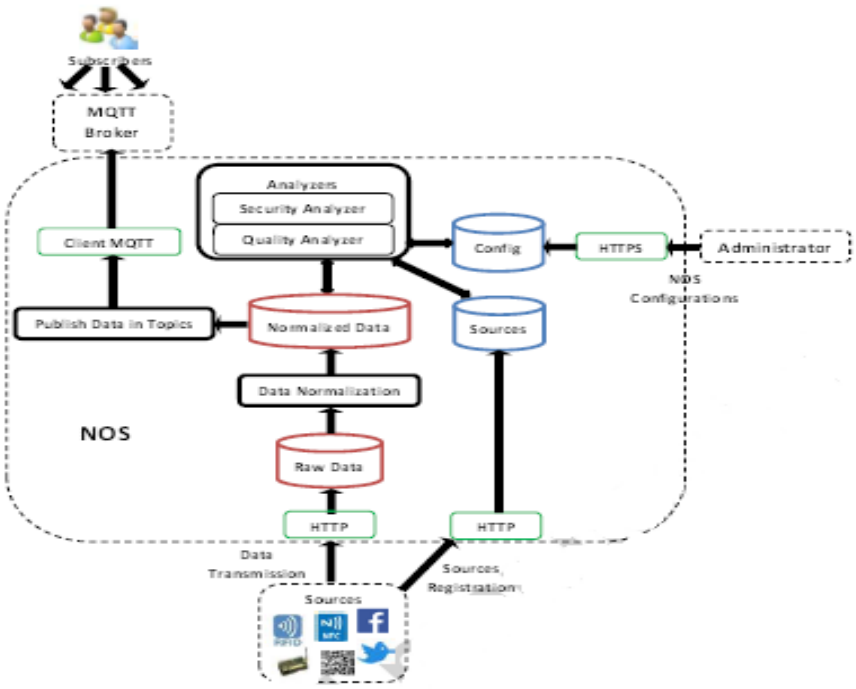

Fig. 1: System Architecture

Two main entities can be identified in an IoT context:(i) the nodes, intended as data sources and represented by heterogeneous devices (e.g., wireless sensor networks, FID, NFC, actuators, social networks); (ii) the users, who access services making use of IoT data through Internet-connected mobile devices (e.g., smart phone, tablet). In our architecture [6], represented in Fig. 1, we also have a layer (the NOS layer) in charge of processing and managing the data generated by the nodes. NOSs are networked smart nodes deployed in a distributed manner and do not present strict constraints in terms of resources and computational capabilities. They are able to interface with nearby IoT devices and to process the acquired data closer to the sources than a centralized platform. In the next sections, through a bottom-up analysis, the components of a NOS are detailed.

The description includes (i) the southbound interfaces (i.e., the ones used to interact with the data sources); (ii) the processing units (in charge of security and data quality evaluation);(iii) the northbound interfaces (i.e., the publish/subscribe mechanism used for the sharing of the information with other services).

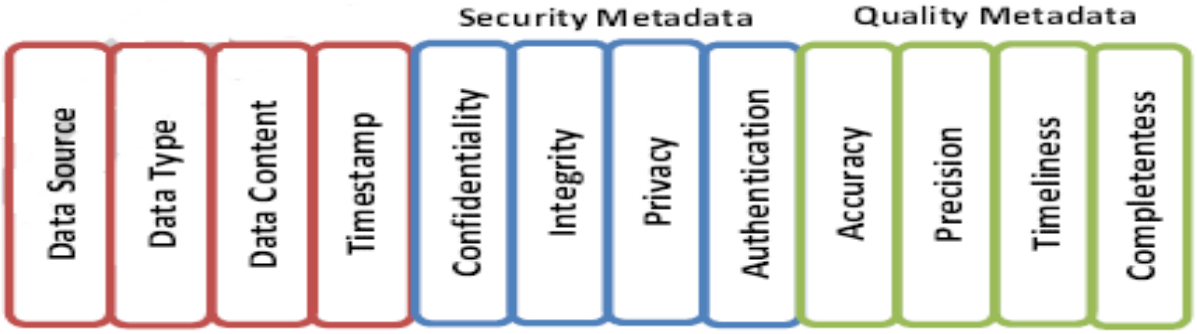

Fig. 2: NOS Data Format

NOSs collect data transmitted by different kinds of sources (i.e., nodes). The system has been designed to support both registered sources as well as anonymous ones, each characterized by different communication technologies and providing diverse quality levels for their data. As regards the registered nodes, NOSs provide a service for source registration by means of HTTP protocol; the related information are stored in the *Sources* data structure. Registered sources are associated with an identifier, and, optionally, with a geographical position and/or an encryption scheme, including the proper keys for interactions with NOSs. For each incoming data NOSs extract the following information: (i) the data source, which describes the type of node (the identifier in case of a registered source); (ii) the communication mode, which is the way in which the data is transmitted(e.g., discrete or streaming communication); (iii) the data schema, which represents the type (e.g., number, text)and the format of the transmitted data; (iv) metadata describing the data content; (v) a timestamp describing when the data was received by the NOS. The HTTP communication protocol is also used among NOS and the data sources for the data transmission. Since received data is highly heterogeneous, NOSs initially stores them in the *Raw Data* collection, and, periodically, elaborates them according to the two-phase process shown in Fig.1 (including *Data Normalization* and

Analyzers) in order to obtain an uniform representation and, as specified in the next section, enriching it with relevant metadata.

1.3 lightweight cryptography

The upcoming ERA of pervasive computing will be characterized by many smart devices that—because of the tight cost constraints inherent in mass deployments have very limited resources in terms of memory, computing power, and battery supply. Here, it's necessary to interpret Moore's law differently:

Rather than a doubling of performance, we see a halving of the price for constant computing power every 18 months. Because many foreseen applications have extremely tight cost constraints—for example, RFID in tetra packs—over time, Moore's law will increasingly enable such applications. Many applications will process sensitive health-monitoring or biometric data, so the demand for cryptographic components that can be efficiently implemented is strong and growing. For such implementations, as well as for ciphers that are particularly suited for this purpose, we use the generic term lightweight cryptography in this article.

Every designer of lightweight cryptography must cope with the trade-offs between security, cost, and performance. It's generally easy to optimize any two of the three design goals—security and cost, security and performance, or cost and performance; however, it is very difficult to optimize all three design goals at once. For example, a secure and high-performance hardware implementation can be achieved by a pipelined, side-channel-resistant architecture, resulting in a high area requirement, and thus high costs. On the other hand, it's possible to design a secure, low-cost hardware implementation with the drawback of limited performance.

In this article, we present a selection of recently published lightweight-cryptography implementations and compare them to state-of-the-art results in their field. This survey covers recent hardware and software implementations of symmetric as well as asymmetric ciphers. We will discuss software and hardware implementations separately, because they have different and sometimes contrary characteristics. For example, bit permutations are virtually free in hardware, whereas in software they can significantly slow down implementations. Also, large substitution tables are often software friendly,

but hardware realizations can be relatively costly. Finally, three valuation metric is different: For software implementations, we compare both RAM and ROM requirements and the required number of clock cycles. For hardware implementations, we focus on the required chip size and the number of clock cycles. We don't compare power consumption for the hardware implementations, because different standard-cell technologies were used and estimates from simulating environments are not accurate. Software implementations let us achieve a rough estimate of power consumption by multiplying the processing time byte average power consumption of the target device Another distinction is between symmetric and asymmetric ciphers, because the latter offer more security functionality and therefore have different application scenarios. Symmetric ciphers serve mainly for message integrity checks, entity authentication, and encryption, whereas asymmetric ciphers additionally provide key-management advantages and non repudiation. Asymmetric ciphers are computationally far more demanding, in both hardware and software. The performance gap on constrained devices such as 8-bit microcontrollers is huge. For example, an optimized asymmetric algorithm such as elliptic-curve cryptography (ECC) performs 100 to1,000 times more slowly than a standard symmetric cipher such as the Advanced Encryption Standard(AES) algorithm, which correlates with a two- to three orders-of-magnitude higher power consumption. Unlike block ciphers, which are well investigated and understood, stream ciphers have received little attention from the scientific community. Although this has recently started to change, we cite stream ciphers only for comparison. (The increasing interest in stream ciphers is apparent in projects such as eStream, within the European Network of Excellence in Cryptography, which aims to foster knowledge about stream ciphers.

2. IoT SECURITY OVERVIEW

The IoT offers connectivity for both human-to-machine and machine-to-machine communications. In the near future, everything is likely to be equipped with small embedded devices which are able to connect to the Internet. Such ability is useful for various domains in our daily life: i.e. from building automation, smart city, and surveillance system to all wearable smart devices. However, the more the IoT devices are deployed, the greater our information system is at risk. Indeed, a non-negligible number

of devices in IoT are vulnerable to security attacks, for example, denial of service and replay attacks, due to their constrained resources and the lack of protection methods. This kind of attacks leads to sensor battery depletion and results in poor performances of sensing applications. In more serious cases, information leak from such tiny devices can expose sensitive data to the outside. In this section, we firstly present the essential security properties for the IoT. Then, we summarize the challenges to be addressed in the IoT.

٢،١ Security properties

Several security properties may need to be satisfied in order to secure the IoT. These general security properties have been also identified in [7,8]. Generally, the security services that should be provided include confidentiality, integrity, authentication, authorization, and freshness. The security requirements are centered on data if sensitive data measured or shared by IoT devices may need to ben protected. Security requirements may also involve controlled access to other resources, for instance the IoT net work layer. Table 1 defines the security properties that will be discussed in this document in relation with the security protocols and solutions proposed for the IoT.

Table 1: Security properties for security protocols in IoT.

Confidentiality	Exchanged messages in the IoT may need to be protected. An attacker should not gain knowledge about the messages exchanged between a sensor node and any other Internet entity
Integrity	The alteration of messages should be detected by the receiver
Authentication	The receiver should be able also to verify the origin of the exchanged messages
Authorization	IoT devices should be able to verify whether certain entities are authorized to access their measured data. At the network layer, only authorized devices should be able to access the IoT network. Unauthorized devices should not be able to route their messages over the IoT devices, because it may deplete their energy
Freshness	This property ensures that no older messages are replayed. This is important to secure the communication channel against replay attacks

٢،٢ Challenges

The heterogonous nature of IoT raises various challenges in terms of data security and network functionality. A secure and operational IoT must overcome the challenges given in Table 2 in order to fulfill the above security requirements.

Table 2: Research Challenges in IoT.

Interoperability Resource Constraints	Deploying security solutions in the IoT should not hinder the functional operation of interconnected heterogeneous devices Most of IoT devices are limited in terms of CPU, memory capacity and battery supply. They often operate on loss and low bandwidth communication channels. It seems to be impossible to apply directly standard conventional security protocols of the Internet in the context of IoT. As an example, the use of small packets (i.e. IEEE 802.15.4 supports only 127-bytes packets [9]) may result in fragmentation of larger packets when using the standard protocols. This will exhaust the life time of sensor nodes and open new possibility of DoS attacks. Hence, the standard security protocols must be redesigned to adapt such difficult scenario, in order to offer equivalent security levels but more efficient performance for the IoT
Availability	The sensor nodes must be available when needed. High availability network of things should remain functional, especially against denial-of-service attacks, such as flooding of incoming messages to targeted nodes forcing them to shut down
Resilience toattacks	The system has to avoid single points of failure so a compromised node will not affect the whole system. Besides, the secured network must also avoid the resource-depletion attacks launched against resource-constrained devices
Privacyprotection	The popularity of RFID tags has raised privacy concerns because anyone can track tags and find the identity of the objects carrying them. In addition, as wearable technology increases its pace, we will be soon able to connect our bodies to the Internet by ‘‘putting on’’ tiny hardware devices (ex. Implant chips inside our bodies). Consequently, our personal information (i.e. healthcare records)must remain secured and should not be traceable, linkable and identifiable
Scalability	The IoT network, for instance WSNs, is generally composed of a large number of devices. The proposed security protocol should beagle to scale. This property is tightly related to the amount of information that each device has to keep in memory for a secure channel to be negotiated with as many entities as possible (other sensor nodes or Internet entities)

٢،٣ Taxonomy of Security Protocols for the IoT

The life cycle of a "thing" is composed of three phases (as denoted in [7]): bootstrapping, operational and maintenance phases. The bootstrapping phase refers to any processing tasks required before the network can operate. Surabaya et al. [8] also define that this process involves a number of settings to be transferred between nodes that shared no prior knowledge of each other. The bootstrap ping step of a device is complete when all security parameters (e.g., secret keys) are securely transferred to the device. This study focuses on recent security solutions proposed for a secure bootstrapping process. The terms and definitions used throughout the rest of the document are presented in Table 3. In this section, we first describe the reference model that illustrates the scenario in which the considered security protocols can be deployed.

Table 3: Abbreviations and Notations

Abbreviation	Definition
IoT	Internet of Things
WSN	Wireless Sensor Network
PKC	Public Key Cryptography
6LBR	6LoWPAN Border Router
DH	Diffie–Hellman exchange
ECC	Elliptic Curve Cryptography
DoS	Denial Of Services
RFID	Radio Frequency Identification

٢,٣,١ Scenario Under Consideration

The security protocols analyzed in this document, as illustrated in Fig. ٣, involve two entities. At least one of them is a device with resource constraints, whereas the second entity can be seen as another constrained device or an external Internet server (i.e., with rich resources). The considered network of "things" consists of a number of tiny nodes communicating with each other and with an unconstrained resource border router (6LBR). The 6LBR is the bridge between the sensor node and the outside world. The 6LBR may take part in the communication between two entities in a passive (transparent to the communicating parties) or active (as a mediator in the communication process) manners. Our study concentrates mainly on securing uncast communications between two entities. Note that group communications are out of scope of this document

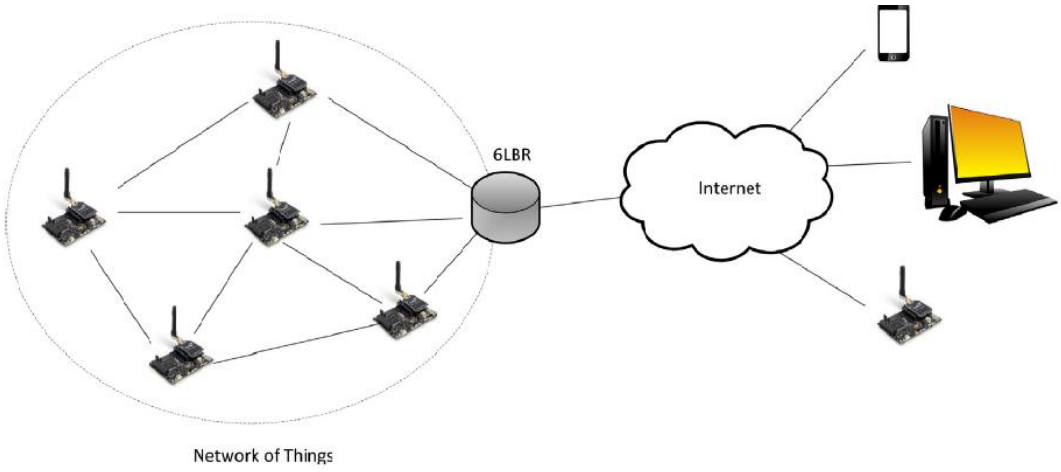

Fig. ٣: Network Architecture of our Scenario

٢,٣,٢ Classification

In this document, existing security solutions for IoT is categorized into two main types: solutions that rely on asymmetric key schemes and solutions that pre-distribute symmetric keys to bootstrap a secure communication. This section describes the two first levels of the proposed taxonomy.

- **Asymmetric key schemes (AKSs)**: The key schemes based on asymmetric cryptography, also known as Public-key cryptography (PKC) are considered as a very common approach to establish a secure communication between two (or more) parties. They employ asymmetric algorithms and are widely deployed in the conventional Internet. The applicability of AKSs in the IoT has one major inconvenience, which is the computation cost and energy consumption. Despite of expensive operations, a lot of researches still seek to apply AKSs in the context of IoT. The proposed approaches can be classified into two categories: key transport based on public key encryption and key agreement based on asymmetric techniques.
- **Key transport based on public key encryption**: Similarly to the traditional key transport mechanism, the first category requires from the public key to securely transport information. Various key establishment techniques have been

proposed for IoT, ranging from raw public key usage to complex implementations in X.509 standard.

- **Key agreement based on asymmetric techniques:** The second category is based on asymmetric primitives in which a shared secret is derived among two or more parties. In this category, we notice obviously the DH protocol [10] and its variants as we will mention later.
- **Symmetric key pre-distribution schemes**: In addition to asymmetric approaches, researchers propose also multiple techniques using symmetric key establishment mechanisms to bootstrap secure communication in the IoT. Symmetric approaches often assume that nodes involved in the key establishment share common credentials. The pre-shared credentials might be a symmetric key or some random bytes flashed into the sensor before its deployment. This category can be divided into two main sub-categories:
- **Probabilistic key distribution**: This sub-category concerns the mechanisms that distribute security credentials (keys, random bytes) chosen randomly from a key pool to constrained nodes. During their initial communication, each two nodes may discover a common key, with certain probability, to establish a secure communication.
- **Deterministic key distribution**: In this sub-category, a deterministic design is applied to create the key pool and to distribute uniformly the keys such that each two nodes share a common key.

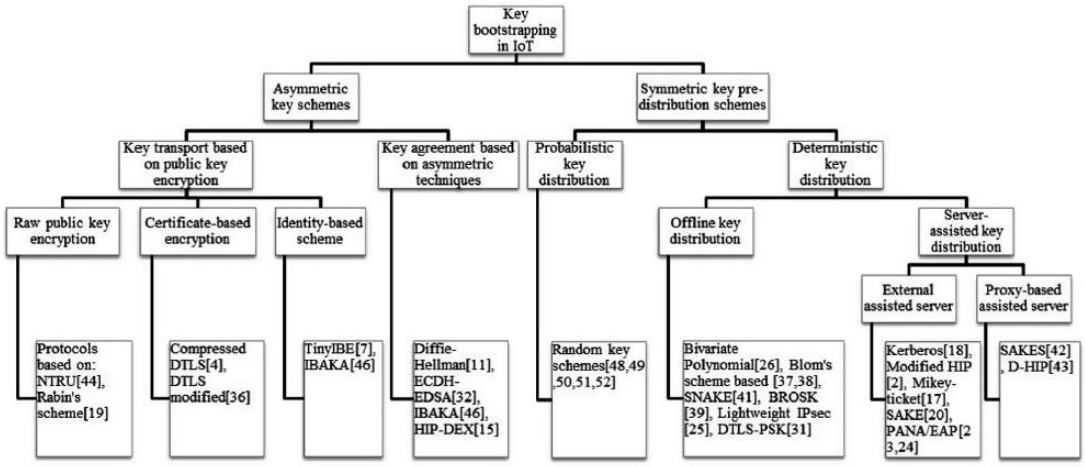
Key bootstrapping in IoT
Asymmetric key schemes
Symmetric key pre-distribution schemes
Key transport based on public key encryption
Key agreement based on asymmetric techniques
Probabilistic key distribution
Deterministic key distribution
Raw public key encryption
Certificate-based encryption
Identity-based scheme
Offline key distribution
Server-assisted key distribution
External assisted server
Proxy-based assisted server
Protocols based on: NTRU[44], Rabin's scheme[19]
Compressed DTLS[4], DTLS modified[36]
TinyIBE[7], IBAKA[46]
Diffie-Hellman[11], ECDH-EDSA[32], IBAKA[46], HIP-DEX[15]
Random key schemes[48,49,50,51,52]
Bivariate Polynomial[26], Blom's scheme based [37,38], SNAKE[41], BROSK [39], Lightweight IPsec [25], DTLS-PSK[31]
Kerberos[18], Modified HIP [2], Mikey-ticket[17], SAKE[20], PANA/EAP[23,24]
SAKES[42], D-HIP[43]

Fig. ٤: Classification of key Bootstrapping Mechanisms in IoT.

3. PROBLEM STATEMENT

This study aims at overcome the challenges of IoT in terms of security which are raised due to the increasing diffusion of services and the heterogonous nature of IoT through provide a lightweight security techniques.

4. OBJECTIVES OF THE STUDY

The objective of this research is to get understanding for the area of Internet of Things (IoT) and lightweight cryptography and contribute new security techniques for betterment of the field. We have found the utilization with these security techniques for IoT and updated ourselves with the latest improvement and related issues. In the following list of the study objectives*:*

- To provide an overview of the emerging Internet of Things (IoT) and related security research work done in this area.
- To present an optimal solutions for different security challenges of IoT in terms of practical use.
- To design and implement a new techniques to secure IoT using lightweight cryptography algorithms.

- To evaluate and analysis the performance of the developing techniques in order to clarify their efficiency and effectiveness in comparing them to some previous techniques that have been done in IoT security.
- To explore various techniques of lightweight cryptography are applied to improve the IoT security.

5. METHODOLOGY

- Analyzing and evaluating the different existing techniques and algorithms that are related to lightweight cryptography and IoT security.
- Using lightweight cryptography concept to produce good security system for IoT.
- Developing a general security framework, where a security techniques balancing between cost, security, and performance to satisfy the requirement of lightweight cryptography.
- Lightweight cryptography is a simple yet efficient way to introduce lightweight security systems.
- Using any simulator in order to build a new technique/algorithm that could help on improving the security of IoT.
- Using any simulator/tools for testing and validating the proposed technique/algorithm.

6. SIGNIFICANCE OF STUDY

This research will be great contribution to our nation in general, in particular is undertaking an ambitious mega project to provide user or organization based systems software to secure their data from attack through smart devices.

This study interest to provide a good solutions to various risks associated to security which are raised due to increasing diffusion of services enabled by IoT technologies. In addition, it is also aims at providing lightweight security systems in order to minimizing data functionality and saving the time, cost of data processing to be proper systems for IoT environment.

After achieving the goal of this research we will be capable to use the power of the IoT security in user or organization based systems software's secure their data from attack.

7. PROPOSED WORK PLAN

Year-wise plan of work and targets to be achieved

Year 1:

- Literature Survey.
- Theoretical analysis of the existing Techniques of lightweight cryptography is applied to improve IoT security.

Year 2:

- Design and implement a new technique to secure IoT.
- Describes an application of presented a new kind of lightweight cryptography and IoT.
- Assessment of experimental results in terms of recognition rate.

Year 3:

- Interpretation of the experimental result.
- Preparation of final synopsis and thesis.
- Submission of synopsis and thesis.

8. REFERENCES

[1] D. Miorandi, S. Sicari, F. De Pellegrini, and I. Chlamtac, "Internetof Things: Vision, applications and research challenges," *Ad HocNetworks*, vol. 10, no. 7, pp. 1497–1516, 2012.

[2] S. Sicari, A. Rizzardi, L. A. Grieco, and A. CoenPorisini, "Security, privacy and trust in Internet of Things: The road ahead," *Computer Networks*, vol. 76, pp. 146–164, 2015.

[3] A. CoenPorisini, P. Colombo, and S. Sicari, *Privacy aware systems: from models to patterns*, igi global ed., 2011.[4] S. Sicari, A. Rizzardi, C. Cappiello, and A. Coen Porisini, "ANFP model for internet of things applications," in *Proc. of IEEEWiMob*, Larnaca, Cyprus, Oct 2014, pp. 164–171.

[4] S. Sicari, A. Rizzardi, C. Cappiello, and A. CoenPorisini, "A NFP model for internet of things applications," in Proc. of IEEE WiMob, Larnaca, Cyprus, Oct 2014, pp. 164–171.

[5] A. Rizzardi, D. Miorandi, S. Sicari, C. Cappiello, and A. CoenPorisini, "Networked smart objects: Moving data processing closer to the source," in *2nd EAI International Conference on IoT as a Service*, Oct 2015.

[6] S. Sicari, C. Cappiello, F. D. Pellegrini, D. Miorandi, and A. CoenPorisini, "A security-and quality-aware system architecture for Internet of Things," *Information Systems Frontiers*, pp. 1–13, 2014.

[7] O. GarciaMorchon, S. Kumar, Security Consideration in the IP-based Internet of Things, CoRE, Internet-draft, 2013.

[8] B. Sarikaya, Y. Ohba, et al., Security Bootstrapping Solutions for Resource-Constrained Devices, Internet-draft, 2012.

[9] S. Raza, S. Duquennoy, T. Chung, et al., Securing communication in 6LoWPAN with compressed IPsec, in: International Conference on Distributed Computing in Sensor Systems and Workshop, 2011.

[10] E. Rescorla, Diffie–Hellman Key Agreement Method, IETF, RFC 2631, 1999.

Research scholar **Research Guide**

Mr.Abdulrazza Hasan Abdullah Al-Ahdal **Dr. Deshmukh N.K.**